Integrating Java and Python
Python for Non-Traditional Programmers

- Ease of Use
- Access To Learning Resources
- Fast, powerful libraries: numpy, scipy
- IPython Notebook
- Great Community
JAVA

- Statically-Typed
- Verbose
- Scales well
- Loved by tool developers and enterprise
- Eclipse RCP, SWT, EMF

PYTHON

- Dynamically-Typed
- Concise
- Ideal for beginners
- Loved by academics and researchers
- Numpy, scipy, matplotlib, IPython notebook
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}

if __name__ == '__main__':
    print "Hello, World!"
Case Study: Diamond LightSource
Scaling Scientific Tools

2005:

"I have all the data I have ever collected on a floppy disk and process it by hand...“

- Principal Beam Line Scientist
- Diamond Light Source Synchrotron

(hat-tip: Mark Basham: Mark Basham - Python for High Throughput Science, PyData 2014)
List of all perspectives and tools available to perform data analysis.

ARPES Data Reduction
DEXPLORE
DIVA
Data Browsing
MX Live Analysis Overview
PEEMA
Python/Python Scripting
(Pre-release) MX pipelines
(Pre-release) NCD Calibration
(Pre-release) NCD Data Reduction
(Pre-release) Workflows
(Pre-release) XAFS
Requirements

- Control the User Interface

- Python to Java
  - In Python, being able to access Java objects
  - Moving data around
  - Needed for Plotting arrays in the workbench
  - (while still running your existing scripts)

- Java to Python
  - In Java, code can access scripts in Python
  - Workflows, Extending Eclipse with Python
1. PyDev

- C Python
- Jython
- PyPy
- IronPython
- Cython
- IPython
1. PyDev - Debug

```

def my_sample_function():
    a = 1
    b = 2
    c = (a, b)
    d = (c, 3)
    print(a, b, c, d)

my_sample_function()
```
1. PyDev - Integration
2. Jython

- Java implementation of Python
- Runs in JVM so can access all JAVA classes

- Jython cannot access C modules
- JyNi could solve this, but
  - Only initial numpy support
  - No Windows support
  - Licensing issues
- Separate environment
3. Py4J

- “A Bridge between Java and Python”
- Runs in JVM and allows Python to access JAVA classes
- Enables Python to Java
- Allows control of User Interface and ability to run your scripts

- Creates proxies rather than moving data
- Exposes everything
- No Java to Python support
3. Py4J

- Interactive autocompletion of Java code in Python
- Autogenerated pop-up help based on method signatures
4. Analysis RPC

Custom solution to move data around

Enhanced Remote Procedure Call

- Datasets (ndarrays)
- Regions Of Interest (ROIs)
- Exceptions

Language Independent
Server and client written for Java and Python
Python

```python
x = numpy.ndarray(...)  
y = numpy.ndarray(...)  
_rpcclient.Plotter.line( x, y)
```

Java

```java
import org.eclipse.dawnsci.IDataset
class Plotter {
public void line(IDataset x, IDataset y) {
    // ...
}
}
```

Transport  
(XML-RPC + Disk/Memory)

Meta and simple data:  
Target → Plotter  
Method → line  
x → /tmp/x  
y → /tmp/y

Datasets:  
/tmp/x  
/tmp/y
Java

```java
interface IFilter {
    IDataset filter(IDataset data);
}

// create client for PyFilter
client.newProxyInstance(IFilter.class)
IDataset result = proxy.filter(data)
```

Python

```python
class PyFilter:
    def filter(data):
        # manipulate ndarray data
        return data

# create server
server.add_handler(PyFilter())
```

Transport

(XML-RPC + Disk/Memory)

- Meta and simple data:
  - Target → PyFilter
  - Method → filter
  - data → /tmp/data

Datasets:

/tmp/data
Python To Java

- Workflows – Python Actor
AnalysisRPC

- Allows moving data around efficiently
- Java to Python, Python to Java
- Generic Calling Infrastructure
- Way of Handling Exceptions

- In DAWNSci today – needs repackaging for use more widely.
- Support for custom datatypes
Java & Python Integration is possible thanks to a mix of different technologies:

- PyDev
- Jython
- AnalysisRPC
- Py4J

Going forward can we unify this technology?
Eclipse Advanced Scripting Environment (EASE)

```
// use help('<topic>') to get more information
new Packages.org.eclipse.ease.modules.EnvironmentModule().loadModule("/System/Env:
 org.eclipse.ease.modules.EnvironmentModule85f1267c9

loadModule("/System/UI")
showInfo001ialog("Hello world", 'Greetings');
```
1. Integrate Py4J into EASE
2. Integrate AnalysisRPC into EASE
3. Integrate EASE (now with Py4J and AnalysisRPC) into SWG projects

Scope?
- Support for additional data structures
- Annotations, code generation
1. Integrate Py4J into EASE

- Py4JPythonScriptEngine
  - Threading model
  - Memory Management
- User Experience
  - Integrate best of PyDev
- What about EASE into Eclipse Platform?
Python & Java

With tight integration Java & Python work together to create a powerful environment for Science, and beyond!

Keep upto date with latest at science working group forum

science.eclipse.org
Questions?

- Tracy Miranda  tracy@kichwacoders.com
- Jonah Graham